Final Project draft - Part 1: Classification

Jacob Walsh

December 4, 2016

library(knitr)  
opts\_chunk$set(tidy.opts=list(width.cutoff=65),tidy=TRUE, cache=TRUE)

charity <- read.csv("C:/Penn State MAS/Stat 897D/charity.csv")  
# transform data  
t.charity = charity  
t.charity$avhv = log(charity$avhv)  
t.charity$incm = log(charity$incm)  
t.charity$inca = log(charity$inca)  
t.charity$plow = charity$plow^(1/3)  
t.charity$tgif = log(charity$tgif)  
t.charity$lgif = log(charity$lgif)  
t.charity$rgif = log(charity$rgif)  
t.charity$tdon = log(charity$tdon)  
t.charity$tlag = log(charity$tlag)  
t.charity$agif = log(charity$agif)  
  
# partition data  
data.train <- t.charity[t.charity$part == "train", ]  
x.train <- data.train[, 2:21]  
c.train <- data.train[, 22] # donr  
n.train.c <- length(c.train) # 3984  
y.train <- data.train[c.train == 1, 23] # damt for observations with donr=1  
n.train.y <- length(y.train)  
  
  
data.valid <- t.charity[t.charity$part == "valid", ]  
x.valid <- data.valid[, 2:21]  
c.valid <- data.valid[, 22] # donr  
n.valid.c <- length(c.valid) # 2018  
y.valid <- data.valid[c.valid == 1, 23] # damt for observations with donr=1  
n.valid.y <- length(y.valid)  
  
  
data.test <- t.charity[t.charity$part == "test", ]  
n.test <- dim(data.test)[1] # 2007  
x.test <- data.test[, 2:21]  
  
x.train.mean <- apply(x.train, 2, mean)  
x.train.sd <- apply(x.train, 2, sd)  
x.train.std <- t((t(x.train) - x.train.mean)/x.train.sd) # standardize to have zero mean and unit sd  
apply(x.train.std, 2, mean) # check zero mean

## reg1 reg2 reg3 reg4 home   
## 2.151811e-17 -2.526099e-17 3.693258e-17 -6.017778e-17 -9.663428e-18   
## chld hinc genf wrat avhv   
## -2.051129e-17 -1.463197e-17 4.465563e-17 -1.062688e-16 -3.616154e-16   
## incm inca plow npro tgif   
## -3.127144e-16 3.759737e-16 2.157653e-16 -6.655491e-17 -2.230764e-16   
## lgif rgif tdon tlag agif   
## -2.126484e-16 -1.180703e-18 -3.788350e-17 2.412638e-17 -1.978122e-16

apply(x.train.std, 2, sd) # check unit sd

## reg1 reg2 reg3 reg4 home chld hinc genf wrat avhv incm inca plow npro tgif   
## 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1   
## lgif rgif tdon tlag agif   
## 1 1 1 1 1

data.train.std.c <- data.frame(x.train.std, donr = c.train) # to classify donr  
data.train.std.y <- data.frame(x.train.std[c.train == 1, ], damt = y.train) # to predict damt when donr=1  
  
x.valid.std <- t((t(x.valid) - x.train.mean)/x.train.sd) # standardize using training mean and sd  
data.valid.std.c <- data.frame(x.valid.std, donr = c.valid) # to classify donr  
data.valid.std.y <- data.frame(x.valid.std[c.valid == 1, ], damt = y.valid) # to predict damt when donr=1  
  
x.test.std <- t((t(x.test) - x.train.mean)/x.train.sd) # standardize using training mean and sd  
data.test.std <- data.frame(x.test.std)

## Subset selection

library(MASS)  
library(leaps)

## Warning: package 'leaps' was built under R version 3.3.1

model.glm = glm(donr ~ ., data.train.std.c, family = "binomial")  
summary(model.glm)

##   
## Call:  
## glm(formula = donr ~ ., family = "binomial", data = data.train.std.c)  
##   
## Deviance Residuals:   
## Min 1Q Median 3Q Max   
## -3.12304 -0.44583 0.06131 0.52125 2.85083   
##   
## Coefficients:  
## Estimate Std. Error z value Pr(>|z|)   
## (Intercept) -0.250947 0.050637 -4.956 7.20e-07 \*\*\*  
## reg1 0.560757 0.061902 9.059 < 2e-16 \*\*\*  
## reg2 1.212348 0.070345 17.234 < 2e-16 \*\*\*  
## reg3 0.019819 0.056783 0.349 0.72707   
## reg4 -0.007802 0.058160 -0.134 0.89328   
## home 1.120833 0.069528 16.120 < 2e-16 \*\*\*  
## chld -1.955590 0.067589 -28.934 < 2e-16 \*\*\*  
## hinc 0.094379 0.051788 1.822 0.06839 .   
## genf -0.024081 0.048450 -0.497 0.61917   
## wrat 0.824907 0.056535 14.591 < 2e-16 \*\*\*  
## avhv 0.063672 0.094261 0.675 0.49937   
## incm 0.456694 0.102563 4.453 8.48e-06 \*\*\*  
## inca 0.031301 0.116410 0.269 0.78802   
## plow -0.044970 0.100429 -0.448 0.65432   
## npro 0.087382 0.101229 0.863 0.38802   
## tgif 0.440793 0.101604 4.338 1.44e-05 \*\*\*  
## lgif -0.121388 0.109195 -1.112 0.26628   
## rgif -0.023264 0.094833 -0.245 0.80621   
## tdon -0.129548 0.049556 -2.614 0.00894 \*\*   
## tlag -0.432944 0.050240 -8.617 < 2e-16 \*\*\*  
## agif 0.103309 0.088528 1.167 0.24323   
## ---  
## Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1  
##   
## (Dispersion parameter for binomial family taken to be 1)  
##   
## Null deviance: 5523.0 on 3983 degrees of freedom  
## Residual deviance: 2763.2 on 3963 degrees of freedom  
## AIC: 2805.2  
##   
## Number of Fisher Scoring iterations: 6

# lasso  
library(glmnet)

## Warning: package 'glmnet' was built under R version 3.3.1

## Loading required package: Matrix

## Loading required package: foreach

## Warning: package 'foreach' was built under R version 3.3.1

## Loaded glmnet 2.0-5

grid = 10^seq(10, -2, length = 100)  
xtrain = model.matrix(donr ~ ., data = data.train.std.c)  
ytrain = data.train.std.c$donr  
xtest = model.matrix(donr ~ ., data.valid.std.c)  
ytest = data.valid.std.c$donr  
  
model.lasso = glmnet(xtrain, ytrain, alpha = 1, lambda = grid, family = "binomial")  
cv.lasso = cv.glmnet(xtrain, ytrain, alpha = 1, family = "binomial")  
lambda = cv.lasso$lambda.min  
lasso.pred = predict(model.lasso, s = lambda, newx = xtest)  
mean((lasso.pred - ytest)^2)

## [1] 4.714624

lasso.coef = predict(model.lasso, type = "coefficients", s = lambda)  
lasso.coef

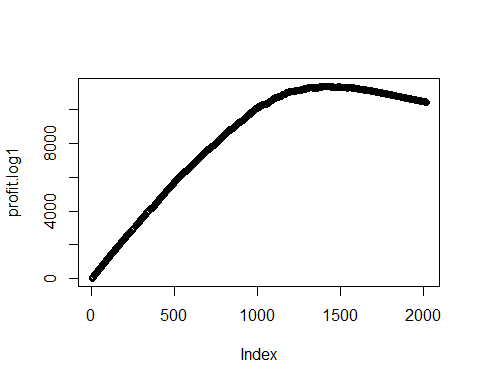
## 22 x 1 sparse Matrix of class "dgCMatrix"  
## 1  
## (Intercept) -0.16010729  
## (Intercept) .   
## reg1 0.36574564  
## reg2 0.91177464  
## reg3 .   
## reg4 .   
## home 0.83866946  
## chld -1.60233044  
## hinc .   
## genf .   
## wrat 0.62434337  
## avhv 0.01100049  
## incm 0.35897272  
## inca 0.01802050  
## plow -0.03896266  
## npro 0.05126886  
## tgif 0.31841709  
## lgif .   
## rgif .   
## tdon -0.03950951  
## tlag -0.30222126  
## agif .

## Model Comparison

### Logistic

1-Full 2-Subset selected through lasso and signficance of predictors combined with guess and checking to increase profit

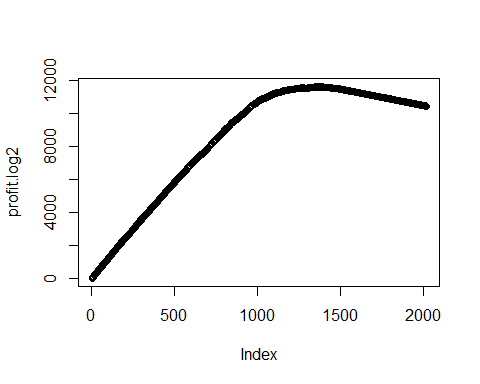
model.log1 = glm(donr ~ ., data = data.train.std.c, family = binomial("logit"))  
post.valid.log1 = predict(model.log1, data.valid.std.c, type = "response")  
profit.log1 <- cumsum(14.5 \* c.valid[order(post.valid.log1, decreasing = T)] -   
 2)  
plot(profit.log1) # see how profits change as more mailings are made



n.mail.valid1 <- which.max(profit.log1) # number of mailings that maximizes profits  
c(n.mail.valid1, max(profit.log1))

## [1] 1392.0 11382.5

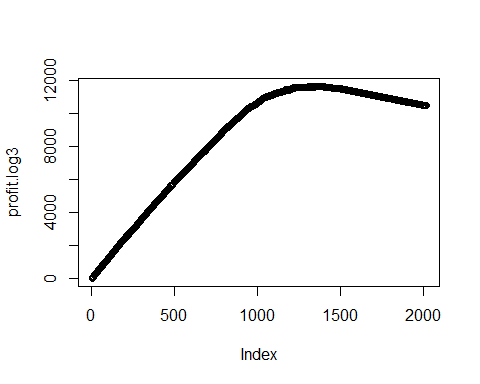
model.log2 = glm(donr ~ reg1 + reg2 + home + chld + wrat + incm +   
 tgif + tdon + tlag + I(hinc^2) + I(tgif^2), data = data.train.std.c,   
 family = binomial("logit"))  
post.valid.log2 = predict(model.log2, data.valid.std.c, type = "response")  
profit.log2 <- cumsum(14.5 \* c.valid[order(post.valid.log2, decreasing = T)] -   
 2)  
plot(profit.log2) # see how profits change as more mailings are made



n.mail.valid2 <- which.max(profit.log2) # number of mailings that maximizes profits  
c(n.mail.valid2, max(profit.log2))

## [1] 1374.0 11650.5

model.log3 = glm(donr ~ . + I(hinc^2), data = data.train.std.c, family = binomial("logit"))  
post.valid.log3 = predict(model.log3, data.valid.std.c, type = "response")  
profit.log3 <- cumsum(14.5 \* c.valid[order(post.valid.log3, decreasing = T)] -   
 2)  
plot(profit.log3) # see how profits change as more mailings are made

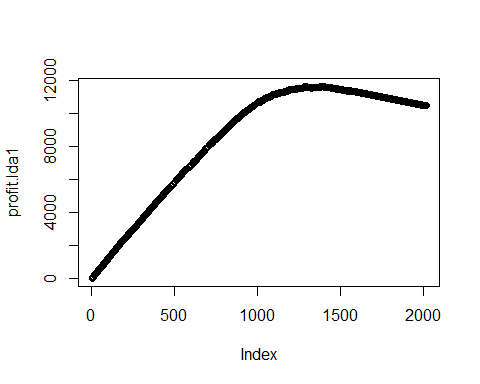


n.mail.valid3 <- which.max(profit.log3) # number of mailings that maximizes profits  
c(n.mail.valid3, max(profit.log3))

## [1] 1371.0 11627.5

### LDA 1-Full 2-Subset

library(MASS)  
model.lda1 = lda(donr ~ . + I(hinc^2), data = data.train.std.c)  
post.valid.lda1 <- predict(model.lda1, data.valid.std.c)$posterior[,   
 2]  
  
profit.lda1 <- cumsum(14.5 \* c.valid[order(post.valid.lda1, decreasing = T)] -   
 2)  
plot(profit.lda1) # see how profits change as more mailings are made



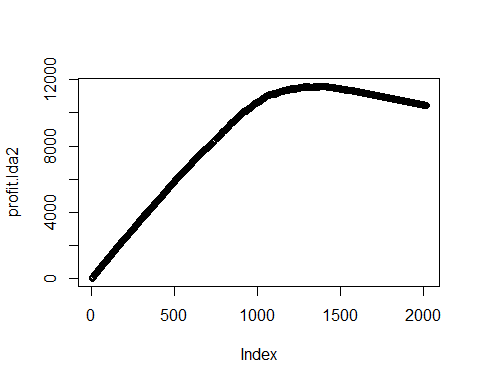
n.mail.valid4 <- which.max(profit.lda1) # number of mailings that maximizes profits  
c(n.mail.valid4, max(profit.lda1)) # report number of mailings and maximum profit

## [1] 1383.0 11632.5

cutoff.lda1 <- sort(post.valid.lda1, decreasing = T)[n.mail.valid4 +   
 1] # set cutoff based on n.mail.valid  
chat.valid.lda1 <- ifelse(post.valid.lda1 > cutoff.lda1, 1, 0) # mail to everyone above the cutoff  
table(chat.valid.lda1, c.valid) # classification table

## c.valid  
## chat.valid.lda1 0 1  
## 0 629 6  
## 1 390 993

model.lda2 = lda(donr ~ reg1 + reg2 + home + chld + wrat + incm +   
 tgif + tdon + tlag + I(hinc^2) + I(tgif^2), data = data.train.std.c)  
  
post.valid.lda2 <- predict(model.lda2, data.valid.std.c)$posterior[,   
 2]  
  
profit.lda2 <- cumsum(14.5 \* c.valid[order(post.valid.lda2, decreasing = T)] -   
 2)  
plot(profit.lda2) # see how profits change as more mailings are made



n.mail.valid5 <- which.max(profit.lda2) # number of mailings that maximizes profits  
c(n.mail.valid5, max(profit.lda2)) # report number of mailings and maximum profit

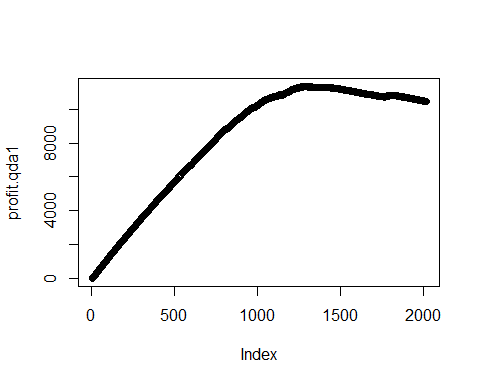
## [1] 1391 11602

cutoff.lda2 <- sort(post.valid.lda2, decreasing = T)[n.mail.valid5 +   
 1] # set cutoff based on n.mail.valid  
chat.valid.lda2 <- ifelse(post.valid.lda2 > cutoff.lda2, 1, 0) # mail to everyone above the cutoff  
table(chat.valid.lda2, c.valid) # classification table

## c.valid  
## chat.valid.lda2 0 1  
## 0 620 7  
## 1 399 992

### QDA 1-Full 2-Subset

model.qda1 = qda(donr ~ ., data = data.train.std.c)  
post.valid.qda1 <- predict(model.qda1, data.valid.std.c)$posterior[,   
 2]  
  
profit.qda1 <- cumsum(14.5 \* c.valid[order(post.valid.qda1, decreasing = T)] -   
 2)  
plot(profit.qda1) # see how profits change as more mailings are made



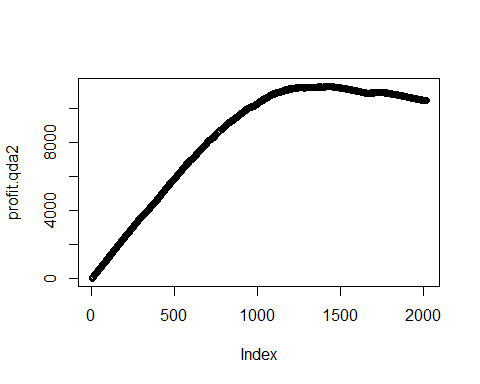
n.mail.valid6 <- which.max(profit.qda1) # number of mailings that maximizes profits  
c(n.mail.valid6, max(profit.qda1)) # report number of mailings and maximum profit

## [1] 1272 11347

cutoff.qda1 <- sort(post.valid.qda1, decreasing = T)[n.mail.valid6 +   
 1] # set cutoff based on n.mail.valid  
chat.valid.qda1 <- ifelse(post.valid.qda1 > cutoff.qda1, 1, 0) # mail to everyone above the cutoff  
table(chat.valid.qda1, c.valid) # classification table

## c.valid  
## chat.valid.qda1 0 1  
## 0 705 41  
## 1 314 958

model.qda2 = qda(donr ~ reg1 + reg2 + home + chld + wrat + incm +   
 tgif + tdon + tlag + I(hinc^2) + I(tgif^2), data = data.train.std.c)  
post.valid.qda2 <- predict(model.qda2, data.valid.std.c)$posterior[,   
 2]  
  
profit.qda2 <- cumsum(14.5 \* c.valid[order(post.valid.qda2, decreasing = T)] -   
 2)  
plot(profit.qda2) # see how profits change as more mailings are made



n.mail.valid7 <- which.max(profit.qda2) # number of mailings that maximizes profits  
c(n.mail.valid7, max(profit.qda2)) # report number of mailings and maximum profit

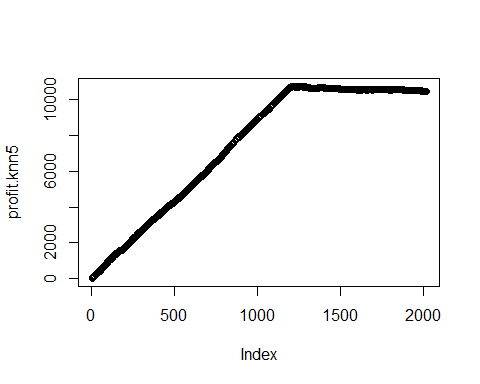
## [1] 1423 11306

cutoff.qda2 <- sort(post.valid.qda2, decreasing = T)[n.mail.valid7 +   
 1] # set cutoff based on n.mail.valid  
chat.valid.qda2 <- ifelse(post.valid.qda2 > cutoff.qda2, 1, 0) # mail to everyone above the cutoff  
table(chat.valid.qda2, c.valid) # classification table

## c.valid  
## chat.valid.qda2 0 1  
## 0 572 23  
## 1 447 976

### KNN

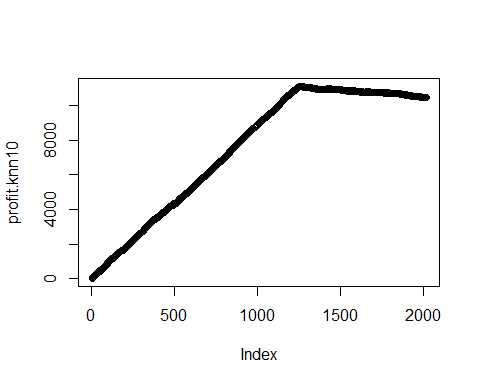
library(class)  
set.seed(1)  
post.valid.knn5 = knn(x.train.std, x.valid.std, c.train, k = 5)  
profit.knn5 <- cumsum(14.5 \* c.valid[order(post.valid.knn5, decreasing = T)] -   
 2)  
plot(profit.knn5)



n.mail.valid8 = which.max(profit.knn5)  
c(n.mail.valid8, max(profit.knn5))

## [1] 1213 10740

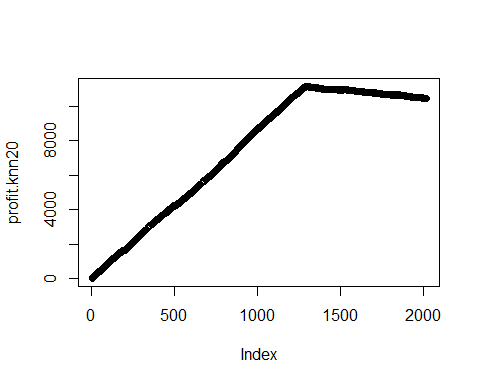
set.seed(1)  
post.valid.knn10 = knn(x.train.std, x.valid.std, c.train, k = 10)  
profit.knn10 <- cumsum(14.5 \* c.valid[order(post.valid.knn10, decreasing = T)] -   
 2)  
plot(profit.knn10)



n.mail.valid9 = which.max(profit.knn10)  
c(n.mail.valid9, max(profit.knn10))

## [1] 1258 11114

set.seed(1)  
post.valid.knn20 = knn(x.train.std, x.valid.std, c.train, k = 20)  
profit.knn20 <- cumsum(14.5 \* c.valid[order(post.valid.knn20, decreasing = T)] -   
 2)  
plot(profit.knn20)



n.mail.valid10 = which.max(profit.knn20)  
c(n.mail.valid10, max(profit.knn20))

## [1] 1295 11156

### GAM

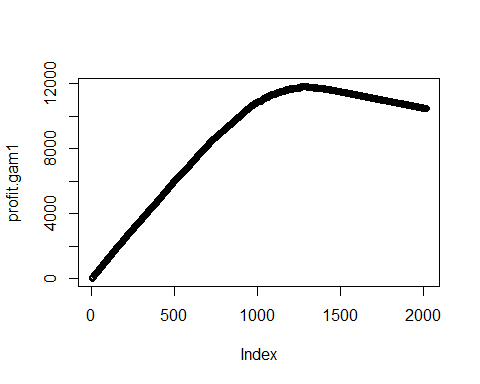
library(gam)

## Warning: package 'gam' was built under R version 3.3.1

## Loading required package: splines

## Loaded gam 1.14

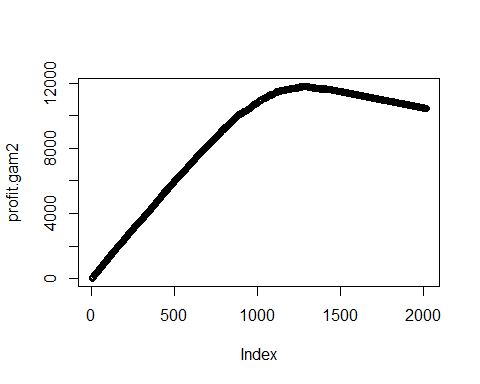
model.gam1 = gam(donr ~ reg1 + reg2 + home + chld + wrat + avhv +   
 incm + tlag + tgif + s(tdon, df = 3) + s(hinc, df = 5), data = data.train.std.c,   
 family = binomial)  
  
post.valid.gam1 <- predict(model.gam1, data.valid.std.c, type = "response")  
profit.gam1 <- cumsum(14.5 \* c.valid[order(post.valid.gam1, decreasing = T)] -   
 2)  
plot(profit.gam1)



n.mail.valid11 = which.max(profit.gam1)  
c(n.mail.valid11, max(profit.gam1))

## [1] 1273.0 11823.5

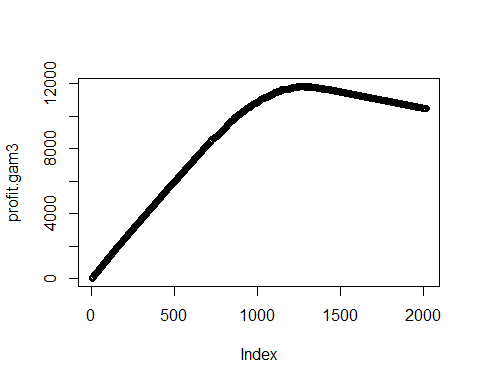
model.gam2 = gam(donr ~ reg1 + reg2 + home + chld + wrat + s(hinc,   
 df = 3) + s(incm, df = 3) + s(tgif, df = 3) + s(tdon, df = 3) +   
 s(tlag, df = 3), data = data.train.std.c, family = binomial)  
  
post.valid.gam2 <- predict(model.gam2, data.valid.std.c, type = "response")  
profit.gam2 <- cumsum(14.5 \* c.valid[order(post.valid.gam2, decreasing = T)] -   
 2)  
plot(profit.gam2)



n.mail.valid12 = which.max(profit.gam2)  
c(n.mail.valid12, max(profit.gam2))

## [1] 1280.0 11809.5

model.gam3 = gam(donr ~ reg1 + reg2 + home + chld + wrat + s(hinc,   
 df = 5) + s(incm, df = 5) + s(tgif, df = 5) + s(tdon, df = 5) +   
 s(tlag, df = 5), data = data.train.std.c, family = binomial)  
  
post.valid.gam3 <- predict(model.gam3, data.valid.std.c, type = "response")  
profit.gam3 <- cumsum(14.5 \* c.valid[order(post.valid.gam3, decreasing = T)] -   
 2)  
plot(profit.gam3)



n.mail.valid13 = which.max(profit.gam3)  
c(n.mail.valid13, max(profit.gam3))

## [1] 1261 11833

# having df of 5 across the board increases the profit but at a  
# risk of overfitting, there isn't very much difference between  
# gam3 and gam1

### SVM

library(e1071)

## Warning: package 'e1071' was built under R version 3.3.2

# model.svm1=tune(svm, donr~., data=data.train.std.c, ranges =  
# list(cost = c(0.01, 0.1, 1, 10)), kernel = 'linear')  
  
## tried svm, but took so long that is is impractical.

### Trees

library(caret)  
set.seed(1)  
gbmfit <- train(as.factor(donr) ~ ., data = data.train.std.c, method = "gbm")

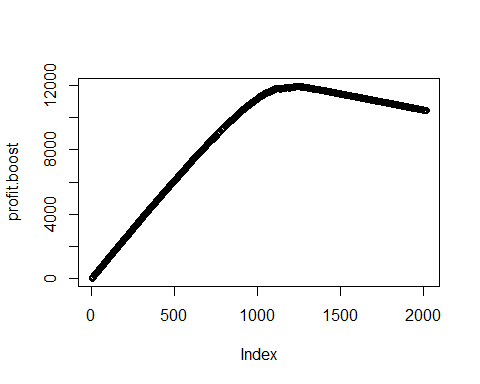
## Optimal GBM

Using the package caret the parameters for the gradient boosting model were tuned to the following:

gbmfit

## Stochastic Gradient Boosting   
##   
## 3984 samples  
## 20 predictor  
## 2 classes: '0', '1'   
##   
## No pre-processing  
## Resampling: Bootstrapped (25 reps)   
## Summary of sample sizes: 3984, 3984, 3984, 3984, 3984, 3984, ...   
## Resampling results across tuning parameters:  
##   
## interaction.depth n.trees Accuracy Kappa   
## 1 50 0.8481621 0.6963938  
## 1 100 0.8780111 0.7560155  
## 1 150 0.8942962 0.7885667  
## 2 50 0.8749656 0.7499185  
## 2 100 0.8992321 0.7984345  
## 2 150 0.9049084 0.8097885  
## 3 50 0.8874692 0.7749204  
## 3 100 0.9032867 0.8065444  
## 3 150 0.9053052 0.8105804  
##   
## Tuning parameter 'shrinkage' was held constant at a value of 0.1  
##   
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10  
## Accuracy was used to select the optimal model using the largest value.  
## The final values used for the model were n.trees = 150,  
## interaction.depth = 3, shrinkage = 0.1 and n.minobsinnode = 10.

set.seed(1)  
boost.mod = gbm(donr ~ ., data = data.train.std.c, distribution = "bernoulli",   
 n.trees = 150, interaction.depth = 3, shrinkage = 0.1, n.minobsinnode = 10)  
  
post.valid.boost <- predict(boost.mod, data.valid.std.c, type = "response",   
 n.trees = 150)  
profit.boost <- cumsum(14.5 \* c.valid[order(post.valid.boost, decreasing = T)] -   
 2)  
plot(profit.boost)



n.mail.valid14 = which.max(profit.boost)  
c(n.mail.valid14, max(profit.boost))

## [1] 1236.0 11955.5

## Summary

# model 1 - full model w/all predictors model 2-subset model  
# w/reg1+reg2+home+chld+wrat+incm+tgif+tdon+tlag  
# +I(hinc^2)+I(tgif^2)  
c(n.mail.valid1, max(profit.log1))

## [1] 1392.0 11382.5

c(n.mail.valid2, max(profit.log2))

## [1] 1374.0 11650.5

c(n.mail.valid3, max(profit.log3)) #full model + hinc^2

## [1] 1371.0 11627.5

c(n.mail.valid4, max(profit.lda1)) #full model + hinc^2

## [1] 1383.0 11632.5

c(n.mail.valid5, max(profit.lda2))

## [1] 1391 11602

c(n.mail.valid6, max(profit.qda1))

## [1] 1272 11347

c(n.mail.valid7, max(profit.qda2))

## [1] 1423 11306

c(n.mail.valid8, max(profit.knn5))

## [1] 1213 10740

c(n.mail.valid9, max(profit.knn10))

## [1] 1258 11114

c(n.mail.valid10, max(profit.knn20))

## [1] 1295 11156

c(n.mail.valid11, max(profit.gam1)) #best model gam(donr~reg1+reg2+home+chld+wrat+avhv+incm+tlag+tgif+s(tdon,df=3)+s(hinc,df=5), data=data.train.std.c, family=binomial)

## [1] 1273.0 11823.5

c(n.mail.valid12, max(profit.gam2))

## [1] 1280.0 11809.5

c(n.mail.valid13, max(profit.gam3))

## [1] 1261 11833

c(n.mail.valid14, max(profit.boost))

## [1] 1236.0 11955.5